

8. isin() method

Used to filter a DataFrame or a Series based on whether each element is present in a list of values. It is particularly useful when you want to filter rows or columns that match any of the values in a specified list.

It returns a Boolean series that can be used for Boolean Indexing.

Syntax:

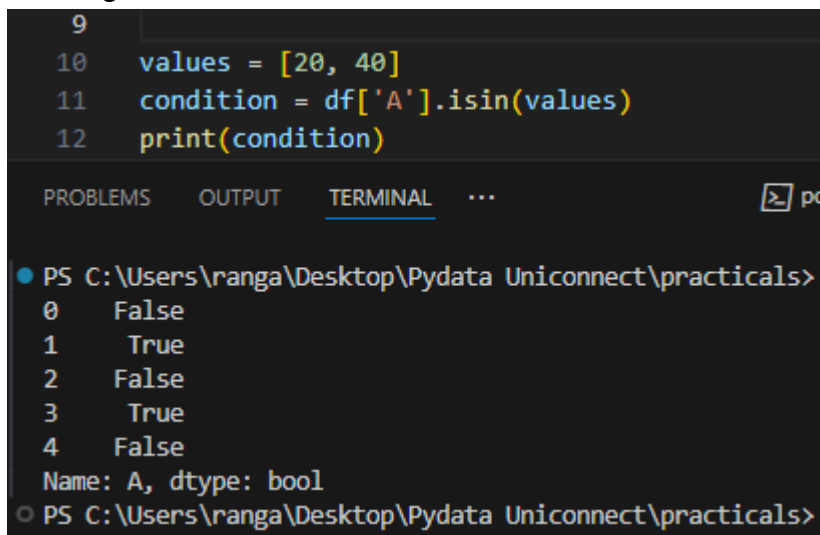
```
DataFrameName.isin(values)
SeriesName.isin(values)
```

- `values` : A list, set or other iterable containing the values to check for.

Ex: Filtering a DataFrame

```
import pandas as pd
data = {
    'A': [10, 20, 30, 40, 50],
    'B': [5, 15, 25, 35, 45]
}
df = pd.DataFrame(data)
```

1. Looking for values 20 and 40 in the Series 'A'



```
9
10 values = [20, 40]
11 condition = df['A'].isin(values)
12 print(condition)
```

PROBLEMS OUTPUT TERMINAL ...

```
● PS C:\Users\ranga\Desktop\Pydata Uniconnect\practicals>
0    False
1     True
2    False
3     True
4    False
Name: A, dtype: bool
○ PS C:\Users\ranga\Desktop\Pydata Uniconnect\practicals>
```

Then you can use this Boolean Series `condition` to filter the DataFrame `df`

```
10 values = [20, 40]
11 condition = df['A'].isin(values)
12 print(condition)
13
14 print('')
15
16 filtered_df = df[condition]
17 print(filtered_df)
```

PROBLEMS OUTPUT TERMINAL ...

```
● PS C:\Users\ranga\Desktop\Pydata Uniconnect\practicals>
0 False
1 True
2 False
3 True
4 False
Name: A, dtype: bool

   A  B
1 20 15
3 40 35
○ PS C:\Users\ranga\Desktop\Pydata Uniconnect\practicals>
```

- Here `filtered_df` only contains rows with column 'A' has values 20 and 40.

2. Filtering with multiple columns

You can apply `isin()` to multiple columns of a DataFrame to check if the values in any of the specified columns match the given values.

```
12 values = [20, 40]
13 filtered_df = df[df.isin(values).any(axis=1)]
14 print(filtered_df)
```

PROBLEMS OUTPUT TERMINAL ...

```
● PS C:\Users\ranga\Desktop\Pydata Uniconnect\practicals>
   A  B
1 20 15
3 40 35
○ PS C:\Users\ranga\Desktop\Pydata Uniconnect\practicals>
```

- Here `any(axis=1)` is used to apply this condition to any column in the DataFrame.
- `axis=1` : Y axis, `axis=0` : X axis

Ex: Filtering a Series

```
import pandas as pd
s = pd.Series([10, 20, 30, 40, 50])
```

```
12 values = [20, 40]
13 filtered_s = s[s.isin(values)]
14 print(filtered_s)
```

PROBLEMS OUTPUT TERMINAL ...



```
● PS C:\Users\ranga\Desktop\Pydata Uniconnect\practicals>
1    20
3    40
dtype: int64
○ PS C:\Users\ranga\Desktop\Pydata Uniconnect\practicals>
```